

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/012772

International filing date: 11 July 2005 (11.07.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-204313
Filing date: 12 July 2004 (12.07.2004)

Date of receipt at the International Bureau: 05 August 2005 (05.08.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日本国特許庁
JAPAN PATENT OFFICE

14.7.2005

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2004年 7月12日

出願番号
Application Number: 特願2004-204313

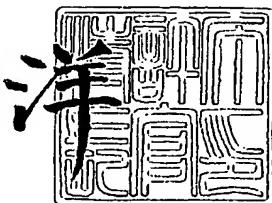
パリ条約による外国への出願
に用いる優先権の主張の基礎
となる出願の国コードと出願
番号
The country code and number
of your priority application,
to be used for filing abroad
under the Paris Convention, is

出願人
Applicant(s): ソニー株式会社

2005年 5月10日

特許庁長官
Commissioner,
Japan Patent Office

小川



出証番号 出証特2005-3040591

【書類名】 特許願
【整理番号】 0490516601
【提出日】 平成16年 7月12日
【あて先】 特許庁長官殿
【国際特許分類】 G06C 1/00
【発明者】
【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社内
【氏名】 鈴木 輝彦
【発明者】
【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社内
【氏名】 石谷 和博
【発明者】
【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社内
【氏名】 矢ヶ崎 陽一
【特許出願人】
【識別番号】 000002185
【氏名又は名称】 ソニー株式会社
【代理人】
【識別番号】 100094053
【弁理士】
【氏名又は名称】 佐藤 隆久
【手数料の表示】
【予納台帳番号】 014890
【納付金額】 16,000円
【提出物件の目録】
【物件名】 特許請求の範囲 1
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【包括委任状番号】 9707389

【書類名】特許請求の範囲

【請求項 1】

直交変換のブロックサイズを示すフラグに応じ、第1のブロックサイズのデータを可変長符号化する場合に、前記第1のブロックサイズのデータを、当該第1のブロックサイズに比べて小さい複数の第2のブロックサイズのデータに分割し、当該分割した第2のブロックサイズについて予め決められたテーブルを用いて当該第2のブロックサイズのデータについて可変長符号化を行う

符号化方法。

【請求項 2】

前記分割した第2のブロックサイズのデータについて前記可変長符号化を行う際に、当該第2のブロックサイズのデータが属する第1のブロックサイズのデータに隣接する他の第1のブロックサイズのデータ内の当該第2のブロックサイズのデータに対応する第2のブロックサイズのデータに用いられた前記テーブルを用いて、前記可変長符号化を行う

請求項1に記載の符号化方法。

【書類名】明細書

【発明の名称】符号化方法

【技術分野】

【0001】

本発明は、MPEG、H.26x等の様に、離散コサイン変換若しくはカールーネン・レーベ変換等の直交変換と動き補償によって圧縮された画像情報（ビットストリーム）を、衛星放送、ケーブルTV、インターネットなどのネットワークメディアを介して受信する際に、若しくは光、磁気ディスク、フラッシュメモリのような記憶メディア上で処理する際に用いられる画像情報符号化方法及び復号化方法に関するものである。

【背景技術】

【0002】

近年、画像情報をデジタルとして取り扱い、その際、効率の高い情報の伝送、蓄積を目的とし、画像情報特有の冗長性を利用して、離散コサイン変換等の直交変換と動き補償により圧縮するMPEGなど的方式に準拠した装置が、放送局などの情報配信、及び一般家庭における情報受信の双方において普及しつつある。

【0003】

特に、MPEG2（ISO/IEC 13818-2）は、汎用画像符号化方式として定義されており、飛び越し走査画像及び順次走査画像の双方、並びに標準解像度画像及び高精細画像を網羅する標準で、プロフェッショナル用途及びコンシューマー用途の広範なアプリケーションに現在広く用いられている。MPEG2圧縮方式を用いることにより、例えば720×480画素を持つ標準解像度の飛び越し走査画像であれば4～8Mbps、1920×1088画素を持つ高解像度の飛び越し走査画像であれば18～22Mbpsの符号量（ビットレート）を割り当てることで、高い圧縮率と良好な画質の実現が可能である。

【0004】

MPEG2は主として放送用に適合する高画質符号化を対象としていたが、MPEG1より低い符号量（ビットレート）、つまりより高い圧縮率の符号化方式には対応していなかった。携帯端末の普及により、今後そのような符号化方式のニーズは高まると思われ、これに対応してMPEG4符号化方式の標準化が行われた。画像符号化方式に関しては、1998年12月にISO/IEC 14496-2としてその規格が国際標準に承認された。

【0005】

更に、近年、当初テレビ会議用の画像符号化を目的として、H.264（ITU-T Q6/16 VCEG）という標準の規格化が進んでいる。H.264はMPEG2やMPEG4といった従来の符号化方式に比べ、その符号化、復号化により多くの演算量が要求されるものの、より高い符号化効率が実現されることが知られている。また、現在、MPEG4の活動の一環として、このH.264をベースに、H.264ではサポートされない機能をも取り入れ、より高い符号化効率を実現する標準化がJoint Mode 1 of Enhanced-Compression Video Codingとして行われている。

【0006】

図1に離散コサイン変換若しくはカールーネン・レーベ変換等の直交変換と動き補償により画像圧縮を実現する画像情報符号化装置のプロック図を示す。すなわち、1はA/D変換装置、2は画面並べ替えバッファ、3は加算器、4は直交変換装置、5は量子化装置、6は可逆符号化装置、7は蓄積バッファ、8は逆量子化装置、9は逆直交変換装置、10はフレームメモリ、11は動き予測・補償装置、12はレート制御装置である。

【0007】

図1において、入力となる画像信号は、まず、A/D変換装置1においてデジタル信号に変換される。次に、出力となる画像圧縮情報のGOP（Group of Pictures）構造に応じ、画面並べ替えバッファ2においてフレームの並べ替えが行われる。

イントラ符号化が行われる画像に関しては、フレーム全体の画像情報が直交変換装置4に入力され、ここで離散コサイン変換、カルーネン・レーベ変換等の直交変換が施される。直交変換係数4の出力となる変換係数は、量子化装置5において量子化処理が施される。量子化装置5の出力となる、量子化された変換係数は、可逆変換装置6に入力され、ここで可変長符号化、算術符号化等の可逆符号化が施された後、蓄積バッファ7に蓄積され、画像圧縮情報として出力される。

量子化装置5の挙動はレート制御装置12によって制御される。同時に、量子化装置5の出力となる、量子化された変換係数は、逆量しか装置8に入力され、更に逆直交変換装置9において逆直交変換処理が施されて、復号化画像情報となり、その情報はフレームメモリ10に蓄積される。

【0008】

インター符号化が行われる画像に関しては、まず、画像情報は動き予測・補償装置11に入力される。同時に参照となる画像情報がフレームメモリ10より取り出され、動き予測・補償処理を施され、参照画像情報が生成される。参照画像情報は加算器3に送られ、ここで当該画像情報との差分信号へ変換される。動き補償・予測装置11は、同時に動きベクトル情報を可逆符号化装置6に出力し、その情報はやはり可変長符号化、算術符号化といった可逆符号化処理が施され、画像圧縮情報のヘッダ部に挿入される情報を形成する。その他の処理はイントラ符号化を施される画像圧縮情報と同様である。

イントラ符号化が行われる画像に関しては、符号化する画素ブロックの近傍の、既に符号化が終了した画素値から予測画像を生成し、その予測画像との差分が符号化される。イントラ符号化された画素は、エンコーダにおいても逆DCT変換、およびそのブロックを符号化する際に使用された予測画像を加算した後、フレームメモリ10に記録される。

イントラ符号化されるブロックでは、既に符号化が終了し、フレームメモリ10に記録されている近傍画素を読み出し、イントラ予測装置14を用いて予測画像を生成する。

イントラ符号化するブロックではイントラ予測装置14において予測画像を生成し、またインター符号化されるブロックに関しては動き予測、動き補償装置11により予測画像を生成する。また、この際に用いられたイントラ予測モードは可変長符号化され、ビットストリームとして伝送される。

【0009】

図2に離散コサイン変換若しくはカルーネン・レーベ変換等の直交変換と動き補償により画像圧縮を実現する画像情報復号化装置のブロック図を示す。すなわち、13は蓄積バッファ、14は可逆符号化装置、15は逆量子化装置、16は逆直交変換装置、17は加算器、18は画面並べ替えバッファ、19は、D/A変換装置、20は動き予測・補償装置、21はフレームメモリ、22はイントラ予測装置である。

【0010】

図2において、入力となる画像圧縮情報は、まず、蓄積バッファ13に格納された後、可逆復号化装置14に転送される。ここで、定められた画像圧縮情報のフォーマットに基づき、可変長復号化、算術復号化等の処理が行われる。同時に、当該フレームがインター符号化されたものである場合には、可逆復号化装置においては、画像圧縮情報のヘッダ部に格納された動きベクトル情報をも復号化し、その情報を動き予測・補償装置20へ転送する。

【0011】

可逆復号化装置の出力となる、量子化された変換係数は、逆量子化装置15に入力され、ここで変換係数として出力される。変換係数は、逆直交変換装置16において、定められた画像圧縮情報のフォーマットに基づき、逆離散コサイン変換、逆カルーネン・レーベ変換等の逆直交変換が施される。当該フレームがイントラ符号化されたものである場合には、逆直交変換処理が施された画像情報は画面並べ替えバッファに格納され、D/A変換処理の後に出力される。

【0012】

当該フレームがインター符号化されたものである場合には、可逆復号化処理が施された

動きベクトル情報、及びフレームメモリ21に格納された画像情報を元に参照画像が生成され、この参照画像と、逆直交変換装置16の出力とが、加算器17において合成される。その他の処理はイントラ符号化されたフレームと同様である。

当該フレームがイントラ符号化されたものである場合には、可変長復号されたイントラ予測モードに従い、フレームメモリから画像を読み出し、イントラ予測装置22により予測画像が生成され、この予測画像と、逆直交変換装置16との出力が加算器7において合成される。

先に述べた Joint Video Team では、MPEG2やMPEG4と同様に動き補償と離散コサイン変換から構成されるハイブリッド符号化方式である。離散コサイン変換の変換方法が 4×4 ブロックサイズの整数係数変換であったり、動き補償の際のブロックサイズが可変であるなど、詳細な方式は異なるが、基本的な方式は図1に示す符号化方式と同様である。

Joint Video Team で標準化が行われている符号化方式（以下JVT Codec）では、MPEG2やMPEG4など既存技術の符号化効率を改善するため、様々な方式が検討されている。例えば、離散コサイン変換の変換方法が 4×4 ブロックサイズの整数係数変換である。また動き補償の際のブロックサイズが可変でありより最適な動き補償が行えるようになっているJVT Codec (H.264|MPEG-4 AVC) ではイントラ予測（フレーム内予測）またはインター予測（フレーム間予測）を行った後、 4×4 ブロックサイズの直交変換を施し、空間的冗長度を削減する。 16×16 画素単位のマクロブロックを 4×4 単位のブロックに分割して直交変換を施す。

当該符号化画素プロックから予測画像を引いた差分信号を $F_{4 \times 4}$ とすると、 4×4 直交変換は以下の通り示すことが出来る。 $f_{4 \times 4}$ が直交変換の出力係数プロックとなる。

【0013】

【数1】

$$f_{4 \times 4} = T_{4 \times 4} \times F_{4 \times 4} \times T_{4 \times 4}^T$$

$$T_{4 \times 4} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}$$

【0014】

整数変換の後、各係数は量子化される。 4×4 ブロックの各成分に関して所定の量子化スケール QP を用いて、量子化される。量子化は例えば以下の式で与えられる。

量子化前の各AC係数の値を f、量子化後の各AC係数の値を f' とすると、量子化後の係数の値は以下の式で表せる。r は丸めのための係数である。

【0015】

【数2】

$$f'[i, j] = (f[i, j] \times Q(QP \% 6, i, j) + r) \gg (15 + QP / 6) \quad (i, j = 0..3)$$

ただし、

$$\begin{aligned} Q(QP \% 6, i, j) &= quantMat[QP \% 6][0], (i, j) \in \{(0,0), (0,2), (2,0), (2,2)\} \\ &= quantMat[QP \% 6][1], (i, j) \in \{(1,1), (1,3), (3,1), (3,3)\} \\ &= quantMat[QP \% 6][2], \text{otherwise} \end{aligned}$$

$$quantMat[6][3] = \begin{bmatrix} 13107 & 5243 & 8224 \\ 11651 & 4660 & 7358 \\ 10486 & 4143 & 6554 \\ 9198 & 3687 & 5825 \\ 8322 & 3290 & 5243 \\ 7384 & 2943 & 4660 \end{bmatrix}$$

【0016】

量子化された変換係数は可変長符号化回路において可変長符号化され、ビットストリームとして出力される。図3に可変長符号化回路の構成を示す。

入力された4x4ブロックの変換係数はスキャン変換回路1により順次読み出される。図4にスキャン変換回路における変換係数の読み出し順をしめす。通常図4(a)のジグザグスキャンの順に係数は読み出され、またフィールド符号化を行う場合、図4(b)のフィールドスキャンの順で係数が読み出される。

スキャン変換回路1により読み出された係数はラン・レベル計算回路2に入力され、ランおよびレベルと呼ばれる情報が計算される。

【0017】

図5にその一例を示す。図5(a)のような4x4直交変換係数の場合、(b)のようなジグザグスキャンの順で読み出され、(c)のような1次元の直交変換係数が読み出される。(スキャン変換回路1)

ラン、レベル計算回路では、この1次元の直交変換係数から、

レベル (level, 係数値)

ラン (run_before, 非0係数値の前の連続する0係数の数)

ランの総和 (total_zeros, 最後の非係数以前の0係数の数)

非0係数の個数 (TotalCoeff)

最後に連続する絶対値1の係数の個数 (TrailingOnes)

最後に連続する絶対値1の係数の符号 (trailing_ones_sign_flag)

【0018】

非0係数の個数 (TotalCoeff) と最後に連続する絶対値1の係数の個数 (TrailingOnes) は2次元可変長符号化回路6に供給され、可変長符号化される。この時、周囲のMBの状態に応じて複数の可変長符号のテーブルの内の一つを適応的に選択して符号化を行う。例えば、左隣りのblockの直交変換係数の個数nAと上隣りのblockの直交変換係数の個数nBから算出される $nC = (nA + nB + 1) \gg 1$ という値によってテーブルの選択が行われる。

表1に実際のTotalCoeffとTrailingOnesの符号化に用いられる可変長符号かのテーブルを示す。例えば、nA=2、nB=3の場合、 $nC = (2+3+1) \gg 1 = 3$ となり、右から4番目のテーブルが使用される。一番右のテーブルの選択基準が $nC = -1$ となっており、これは色差信号のDC値の符号化のみに用いられる。

【0019】

【表 1】

TrailingOnes	TotalCoeff	$0 \leq nC < 2$	$2 \leq nC < 4$	$4 \leq nC < 8$	$8 \leq nC$	$nC = -1$
0	0	1	11	1111	0000 11	01
0	1	0001 01	0010 11	0011 11	0000 00	0001 11
1	1	01	10	1110	0000 01	1
0	2	0000 0111	0001 11	0010 11	0001 00	0001 00
1	2	0001 00	0011 1	0111 1	0001 01	0001 10
2	2	001	011	1101	0001 10	001
0	3	0000 0011 1	0000 111	0010 00	0010 00	0000 11
1	3	0000 0110	0010 10	0110 0	0010 01	0000 011
2	3	0000 101	0010 01	0111 0	0010 10	0000 010
3	3	0001 1	0101	1100	0010 11	0001 01
0	4	0000 0001 11	0000 0111	0001 111	0011 00	0000 10
1	4	0000 0011 0	0001 10	0101 0	0011 01	0000 011
2	4	0000 0101	0001 01	0101 1	0011 10	0000 010
3	4	0000 11	0100	1011	0011 11	0000 000
0	5	0000 0000 111	0000 0100	0001 011	0100 00	-
1	5	0000 0001 10	0000 110	0100 0	0100 01	-
2	5	0000 0010 1	0000 101	0100 1	0100 10	-
3	5	0000 100	0011 0	1010	0100 11	-
0	6	0000 0000 0111 1	0000 0011 1	0001 001	0101 00	-
1	6	0000 0000 110	0000 0110	0011 10	0101 01	-
2	6	0000 0010 01	0000 0101	0011 01	0101 10	-
3	6	0000 0100	0010 0	1001	0101 11	-
0	7	0000 0000 0101 1	0000 0001 111	0001 000	0110 00	-
1	7	0000 0000 0111 0	0000 0011 0	0010 10	0110 01	-
2	7	0000 0000 101	0000 0010 1	0010 01	0110 10	-
3	7	0000 0010 0	0001 00	1000	0110 11	-
0	8	0000 0000 0100 0	0000 0001 011	0000 1111	0111 00	-
1	8	0000 0000 0101 0	0000 0001 110	0001 110	0111 01	-
2	8	0000 0000 0110 1	0000 0001 101	0001 101	0111 10	-
3	8	0000 0001 00	0000 100	0110 1	0111 11	-
0	9	0000 0000 0011 11	0000 0000 1111	0000 1011	1000 00	-
1	9	0000 0000 0011 10	0000 0001 010	0000 1110	1000 01	-
2	9	0000 0000 0100 1	0000 0001 001	0001 010	1000 10	-
3	9	0000 0000 100	0000 010 0	0011 00	1000 11	-
0	10	0000 0000 0010 11	0000 0000 1011	0000 0111 1	1001 00	-
1	10	0000 0000 0010 10	0000 0000 1110	0000 1010	1001 01	-
2	10	0000 0000 0011 01	0000 0000 1101	0000 1101	1001 10	-
3	10	0000 0000 0110 0	0000 0001 100	0001 100	1001 11	-
0	11	0000 0000 0001 111	0000 0000 1000	0000 0101 1	1010 00	-
1	11	0000 0000 0001 110	0000 0000 1010	0000 0111 0	1010 01	-
2	11	0000 0000 0010 01	0000 0000 1001	0000 1001	1010 10	-
3	11	0000 0000 0011 00	0000 0001 000	0000 1100	1010 11	-
1	12	0000 0000 0001 010	0000 0000 1011 0	0000 0101 0	1011 01	-
2	12	0000 0000 0001 101	0000 0000 1101 1	0000 0110 1	1011 10	-
3	12	0000 0000 0010 00	0000 0000 1100	0000 1000	1011 11	-
0	13	0000 0000 0000 1111	0000 0000 0101 1	0000 0011 01	1100 00	-
1	13	0000 0000 0000 001	0000 0000 0101 0	0000 0011 1	1100 01	-
2	13	0000 0000 0001 001	0000 0000 0100 1	0000 0100 1	1100 10	-
3	13	0000 0000 0001 100	0000 0000 0110 0	0000 0110 0	1100 11	-
0	14	0000 0000 0000 1011	0000 0000 0111 1	0000 0101 0	1101 00	-
1	14	0000 0000 0000 1110	0000 0000 0101 11	0000 0011 00	1101 01	-
2	14	0000 0000 0000 1101	0000 0000 0111 0	0000 0101 11	1101 10	-
3	14	0000 0000 0001 000	0000 0000 0100 0	0000 0010 10	1101 11	-
0	15	0000 0000 0000 0111	0000 0000 0101 01	0000 0001 01	1110 00	-
1	15	0000 0000 0000 01010	0000 0000 0010 00	0000 0010 00	1110 01	-
2	15	0000 0000 0000 0001	0000 0000 0010 10	0000 0001 11	1110 10	-
3	15	0000 0000 0000 1100	0000 0000 0000 1	0000 0001 10	1110 11	-
0	16	0000 0000 0000 0100	0000 0000 0001 11	0000 0000 01	1111 00	-
1	16	0000 0000 0000 0110	0000 0000 0001 10	0000 0001 00	1111 01	-
2	16	0000 0000 0000 0101	0000 0000 0001 01	0000 0000 11	1111 10	-
3	16	0000 0000 0000 1000	0000 0000 0001 00	0000 0000 10	1111 11	-

表 1 TrailingOnes と TotalCoeff の VLC テーブル

【0020】

直交変換係数値(level)はレベル符号化回路4に供給されて可変長符号化される。レベル符号化回路4では、直交変換係数値から level_prefix および level suffix と呼ばれるパラメータを抽出する。

level_prefix は以下に示すVCLテーブルを用いて可変長符号化する。

【0021】

【表2】

level_prefix	Bit String
0	1
1	01
2	001
3	0001
4	0000 1
5	0000 01
6	0000 001
7	0000 0001
8	0000 0000 1
9	0000 0000 01
10	0000 0000 001
11	0000 0000 0001
12	0000 0000 0000 1
13	0000 0000 0000 01
14	0000 0000 0000 001
15	0000 0000 0000 0001

表2 level_prefixの復号VLCテーブル

【0022】

level_suffix は suffixLength で与えられるビット長で unsigned integer として符号化される。

level と level_prefix, level_suffix の関係は以下の式で与えられる。

```
levelCode = (level_prefix << suffixLength) + level_suffix
```

levelCodeが偶数の場合 : level = (levelCode + 2) >> 1

levelCodeが偶数でない場合 : level = (-levelCode - 1) >> 1

【0023】

ラン (run_before) およびランの総和 (total_zeros) はラン符号化回路3に供給され可変長符号がされる。以下にはランの総和 (total_zeros)、および、ラン (run_before) に用いられる VLC テーブルを示す。

【0024】

【表3】

total_zeros	TotalCoeff						
	1	2	3	4	5	6	7
0	1	111	0101	00011	0101	000001	000001
1	011	110	111	111	0100	00001	00001
2	010	101	110	0101	0011	111	101
3	0011	100	101	0100	111	110	100
4	0010	011	0100	110	110	101	011
5	00011	0101	0011	101	101	100	11
6	00010	0100	100	100	100	011	010
7	000011	0011	011	0011	011	010	001
8	000010	0010	0010	011	0010	0001	001
9	0000011	00011	00011	0010	00001	001	00000
10	0000010	00010	00010	00010	0001	00000	
11	00000011	000011	000001	00001	00000		
12	00000010	000010	00001	00000			
13	000000011	000001	000000				
14	000000010	000000					
15	000000001						

表3 4x4ブロックのtotal_zeros復号用VLCテーブル (1<=TotalCoeff <=7)

【0025】

【表4】

total_zeros	TotalCoeff							
	8	9	10	11	12	13	14	15
0	000001	000001	00001	0000	0000	000	00	0
1	0001	000000	00000	0001	0001	001	01	1
2	00001	0001	001	001	01	1	1	
3	011	11	11	010	1	01		
4	11	10	10	1	001			
5	10	001	01	011				
6	010	01	0001					
7	001	00001						
8	000000							

表4 4x4ブロックのtotal_zeros復号用VLCテーブル (8<=TotalCoeff <=15)

【0026】

【表5】

total_zeros	TotalCoeff		
	1	2	3
0	1	1	1
1	01	01	0
2	001	00	
3	000		

表5 色差DC 2x2ブロックのtotal_zeros復号用VLCテーブル

【0027】

【表6】

run_before	zerosLeft						
	1	2	3	4	5	6	>6
0	1	1	11	11	11	11	111
1	0	01	10	10	10	000	110
2	-	00	01	01	011	001	101
3	-	-	00	001	010	011	100
4	-	-	-	000	001	010	011
5	-	-	-	-	000	101	010
6	-	-	-	-	-	100	001
7	-	-	-	-	-	-	0001
8	-	-	-	-	-	-	00001
9	-	-	-	-	-	-	000001
10	-	-	-	-	-	-	0000001
11	-	-	-	-	-	-	00000001
12	-	-	-	-	-	-	000000001
13	-	-	-	-	-	-	0000000001
14	-	-	-	-	-	-	00000000001

表6 ラン(run_before)復号用VLCテーブル

【0028】

可変長符号されたこれらの情報は多重化回路5により1本のビットストリームに多重化され出力される。

【発明の開示】

【発明が解決しようとする課題】

【0029】

しかしながら、上記の手法は4x4ブロックサイズでの直交変換にのみ対応しており、8x8ブロックサイズの直交変換など異なるサイズの変換を組み合わせて符号化する、適応的符号化方式には対応できない、という問題がある。

【0030】

本発明においてはこうした事態に対して、8x8, 4x4など異なるブロックサイズの直交変換を行った際に効率よく可変長符号化を実現するものである。

【課題を解決するための手段】

【0031】

上述した従来技術の問題点を解決し、上述した目的を達成するため、本発明においては、直交変換のブロックサイズを示すフラグに応じ、大きいブロックサイズ（例えば8x8ブロック）の場合、小さいブロックサイズ（例えば4x4ブロック）に適応的に再分割し、小さいブロックサイズでの可変長符号化をそれぞれの再分割したブロックに適応することにより、より効率的に可変長符号化を実現することを特徴とする。

【発明の効果】

【0032】

本発明によれば、8x8, 4x4など異なるブロックサイズの直交変換を行った際に効率よく可変長符号化することが可能となる。

【発明を実施するための最良の形態】

【0033】

図6に本発明による画像信号符号化装置の実施例を示す。1はA/D変換装置、2は画面並べ替えバッファ、3は加算器、4は直交変換装置、5は量子化装置、6は可逆符号化装置、7は蓄積バッファ、8は逆量子化装置、9は逆直交変換装置、10はフレームメモリ、11は動き予測・補償装置、12はレート制御装置、13は加算器、14はイントラ予測装置である。

図6において、入力となる画像信号は、まず、A/D変換装置1においてデジタル信号に変換される。次に、ピクチャタイプ決定装置14において決定されるピクチャタイプに応じて、画面並べ替えバッファ2においてフレームの並べ替えが行われる。

【0034】

ピクチャタイプ決定装置14は、各ピクチャの符号化ピクチャタイプを決定する。符号化ピクチャタイプには、イントラ符号化ピクチャ(Iピクチャ)、インター符号化ピクチャ(Pピクチャ、Bピクチャ)がある。各ピクチャをどのび符号化ピクチャタイプで符号化するかは、あらかじめ定められた所定の周期で決定するか、外部からの制御により決定するか、シーンチェンジ検出などを行った結果決定する。またその他の手法でピクチャタイプを決定しても構わない。

【0035】

イントラ符号化が行われる画像に関しては、フレーム全体の画像情報が直交変換装置4に入力され、ここで離散コサイン変換、カルーネン・レーベ変換等の直交変換が施される。直交変換係数4の出力となる変換係数は、量子化装置5において量子化処理が施される。量子化装置5の出力となる、量子化された変換係数は、可逆変換装置6に入力され、ここで可変長符号化、算術符号化等の可逆符号化が施された後、蓄積バッファ7に蓄積され、画像圧縮情報として出力される。量子化装置5の挙動はレート制御装置12によって制御される。同時に、量子化装置5の出力となる、量子化された変換係数は、逆量しき装置8に入力され、更に逆直交変換装置9において逆直交変換処理が施されて、復号化画像情報となり、その情報はフレームメモリ10に蓄積される。

【0036】

インター符号化が行われる画像に関しては、まず、画像情報は動き予測・補償装置11に入力される。同時に参照となる画像情報がフレームメモリ10より取り出され、動き予測・補償処理を施され、参照画像情報が生成される。参照画像情報は加算器3に送られ、ここで当該画像情報との差分信号へ変換される。動き補償・予測装置11は、同時に動きベクトル情報を可逆符号化装置6に出力し、その情報はやはり可変長符号化、算術符号化といった可逆符号化処理が施され、画像圧縮情報のヘッダ部に挿入される情報を形成する。その他の処理はイントラ符号化を施される画像圧縮情報と同様である。

イントラ符号化が行われる画像(Iピクチャ、Iスライス、イントラマクロブロックなど)では、符号化する画素ブロックの近傍の、既に符号化が終了した画素値から予測画像を生成し、その予測画像との差分が符号化される。イントラ符号化された画素は、エンコーダにおいても逆DCT変換、およびそのブロックを符号化する際に使用された予測画像を加算した後、フレームメモリ10に記録される。

【0037】

イントラ符号化されるブロックでは、既に符号化が終了し、フレームメモリ10に記録されている近傍画素を読み出し、イントラ予測装置14を用いて予測画像を生成する。イントラ符号化するブロックではイントラ予測装置14において予測画像を生成し、またインター符号化されるブロックに関しては動き予測、動き補償装置11により予測画像を生成する。また、この際に用いられたイントラ予測モードは可変長符号化され、ビットストリームとして伝送される。

【0038】

本発明においては、直交変換装置4は、例えば、4x4 ブロックサイズおよび8x8 ブロックサイズなど異なるサイズの直交変換を適応的に行う。また、直交変換のサイズに応じて量子化装置5はそれに応じた量子化を行い、また可逆符号化装置6はそれに応じた可逆

符号化を行う。

動き予測・動き補償装置11は既に符号化され、局所復号され、フレームメモリ10に記録されている画像から、動き予測を行い、残差を最小にする動きベクトルおよび動く補償のブロックサイズを決定する。

また、イントラ予測装置14ではイントラ符号化するマクロブロックにおいて、残差が最小となるイントラ予測のモードおよび予測ブロックのサイズを決定する。

直交変換サイズ決定回路では、動き予測・動き補償装置11またはイントラ予測装置14により決定された予測ブロックサイズに基づき、直交変換のブロックサイズを決定し、直交変換のブロックサイズを示す信号 TRSIZE を出力する。

例えば、イントラ予測装置14が 8×8 ブロック単位の予測を行う場合、直交変換のブロックサイズを 8×8 ブロックとする。それ以外の場合、 4×4 ブロック単位の直交変換を行う。

また、例えば、動き予測・動き補償装置11が 8×8 ブロックサイズ以上のブロックサイズで動き補償を行う場合、直交変換のサイズを 8×8 とする。 8×8 ブロックサイズより小さいブロックサイズで動き補償を行う場合、 4×4 ブロックサイズの直交変換を行う。

【0039】

直交変換回路4はTRSIZEに基づき、所定のサイズ（本実施例の場合、 4×4 DCT または 8×8 DCT）の直交変換を行う。

また、量子化装置5は TRSIZEに基づき、実際に適用される直交変換に応じた量子化を行う。例えば、整数精度の直交変換の場合、 4×4 ブロックサイズの変換と 8×8 ブロックサイズの変換では、正規化するための係数が異なる。このため、この正規化の差の分だけスケール値を変更して量子化を行う。

TRSIZE はまた、可逆符号化装置6に入力される。可逆符号化装置6は、これまでの方式では 4×4 直交変換にのみ対応していたため、 4×4 ブロックサイズごとに可逆符号化が行われる。この可逆符号化には表7から表13までの可変長符号化テーブル（VLCテーブル）が用いられる。

これを 8×8 直交変換にも対応するために、本発明では、 8×8 直交変換を行った後、係数を4つの 4×4 ブロックに分割し、この 4×4 サブブロックに対して、表7から12の可変長符号化テーブルを用いて符号化することを特徴とする。これにより、新たに可変長符号化テーブルを用意することなく、適応的に可変長符号化を実現することができる。

【0040】

本発明の実施例における可逆符号化装置6を図7を用いて説明する。TRSIZEが 4×4 直交変換であることを示す場合、可逆符号化装置6は図3と同様に可逆符号化を行う。TRSIZEが 8×8 直交変換であることを示す場合、可逆符号化装置6は図7のような構成で以下に示すような手順に従って符号化される。

入力された 8×8 直交変換の係数はスキャン変換回路1により読み出し順が変更され、例えば図13に示すような順番（ジグザグスキャン）で係数が読み込まれる。このとき(0,0)が一番低い周波数成分である。

スキャン変換回路1から読み出された係数は、サブブロック分割回路6に供給される。このサブブロック分割回路6では、 8×8 ブロックサイズの係数を 4×4 ブロックサイズの係数に再分割する。スキャン変換回路1によりジグザグスキャンの順で読み出された変換係数は、例えば、ジグザグスキャンの順で、先頭から16個の係数ごとに4つの 4×4 サブブロックに分割される。例えば図9に示すように、ジグザグスキャンの順で最初の16個の係数はブロック1の係数であるものとし、次の16個の係数はブロック2の係数であるとする。またその次の16個の係数はブロック3の係数であるものとし、最後の16個の係数はブロック4の係数であるものとする。

4×4 ブロックごとに再分割された係数はランレベル計算回路2に供給され、ランおよびレベルと呼ばれる情報が計算される。

【0041】

ラン、レベル計算回路では、この直交変換係数から、
 レベル (level, 係数値)
 ラン (run_before, 非0係数値の前の連続する0係数の数)
 ランの総和 (total_zeros, 最後の非係数以前の0係数の数)
 非0係数の個数 (TotalCoeff)
 最後に連続する絶対値1の係数の個数 (TrailingOnes)
 最後に連続する絶対値1の係数の符号 (trailing_ones_sign_flag)
 を計算する。

【0042】

非0係数の個数 (TotalCoeff) と最後に連続する絶対値1の係数の個数 (TrailingOnes) は2次元可変長符号化回路7に供給され、可変長符号化される。この時、周囲のMBの状態に応じて複数の可変長符号のテーブルの内の一つを適応的に選択して符号化を行う。例えば、左隣りのblockの直交変換係数の個数nAと上隣りのblockの直交変換係数の個数nBから算出される $nC = (nA+nB+1) \gg 1$ という値によってテーブルの選択が行われる。

本実施例においては TRSIZE が 4×4 直交変換であることを示す場合、上記の方法で nC を計算し、可変長符号化のテーブルの切り替えを行う。また、TRSIZE が 8×8 直交変換であることを示す場合、以下の手順で nC を計算し、可変長符号化のテーブルの切り替えを行う。

【0043】

本実施例を用いて 8×8 直交変換の係数を 4×4 直交変換の係数に再分割した場合、図9のサブブロック1には比較的高い頻度で非ゼロの係数が発生し、ブロック4では逆にほとんどの係数が0となる確率が高くなる。従って、ブロック4などでは値0に短い符号長の符号を割り当てる方が効率が良くまた、ブロック1などでは逆に長い値に短い符号長の符号を割り当てる方が符号化効率が高くなる。ここで、 nC を以下に示すような方式で計算しテーブルを決定する。以下の方程式は一例であり、いずれの方式を用いても構わない。ただし、復号装置でも、符号化装置と組となり同様の方式を用いる必要がある。

【0044】

(方式1)

図8のように現在符号化する 8×8 ブロックをCであるとする。この時、ブロック1は $nC=8$ 、ブロック2および3は $nC=4$ 、ブロック4は $nC=0$ とする。

【0045】

(方式2)

図8のように現在符号化する 8×8 ブロックをCであるとする。この時、ブロック1は $nC=8$ 、ブロック2および3は $nC=2$ 、ブロック4は $nC=0$ とする。

(方式3)

図8のように現在符号化する 8×8 ブロックをCであるとする。この時、ブロック1は $nC=4$ 、ブロック2および3は $nC=2$ 、ブロック4は $nC=0$ とする。

【0046】

(方式4)

図8のように現在符号化する 8×8 ブロックをC、左隣の 8×8 ブロックをA、上隣の 8×8 ブロックをBであるとする。この時、A, Bがともに 8×8 直交変換を用いて符号化されている場合、Cの各サブブロックを符号化する際、A, Bの同じ位置のサブブロックの直交変換の係数の数を用いて nC を計算する。例えば、Aのブロック1の係数の数を nA とし、Bのブロックの係数の数を nB とするとCのブロック1の nC は $nC = (nA+nB+1) \gg 1$ とする。

ブロックAまたはBのいずれかが 8×8 直交変換で、もう一方が 4×4 直交変換である場合、例えば、 8×8 直交変換の同じ位置のブロックの係数の数を nC とする。例えば、ブロックAが 8×8 でブロックBが 4×4 直交変換である場合、Aのブロック1の係数の数 nA がCのブロック1の nC となる。

ブロックAまたはブロックBのいずれも 4×4 直交変換である場合、ブロックCの符号化

しようとしているサブブロックと同じ一のA, Bのサブブロックの数 nA 、 nB から例えば、 $nC = (nA+nB) << 1$ などとして nC を計算する。

上記それぞれの場合の nC の計算方法は一例であり、ブロックAおよびBの情報を用いて別の方法で計算しても構わない。

【0047】

(方式5)

図8のように現在符号化する 8×8 ブロックをC、左隣の 8×8 ブロックをA、上隣の 8×8 ブロックをBであるとする。この時、A, Bがともに 8×8 直交変換を用いて符号化されている場合、Cの各サブブロックを符号化する際、A, Bの同じ位置のサブブロックの直交変換の係数の数を用いて nC を計算する。例えば、Aのブロック1の係数の数を nA とし、Bのブロックの係数の数を nB とするとCのブロック1の nC は $nC = (nA+nB+1) >> 1$ とする。

ブロックAまたはBのいずれかが 8×8 直交変換で、もう一方が 4×4 直交変換である場合、例えば、 8×8 直交変換の同じ位置のブロックの係数の数を nC とする。例えば、ブロックAが 8×8 でブロックBが 4×4 直交変換である場合、Aのブロック1の係数の数 nA がCのブロック1の nC となる。

ブロックAまたはブロックBのいずれも 4×4 直交変換である場合、ブロックCは例えば方式1から3のようにCの符号化するサブブロックの位置に応じて nC を切り替える。

表7に実際のTotalCoeffとTrailingOnesの符号化に用いられる可変長符号かのテーブルを示す。例えば、 $nA=2$ 、 $nB=3$ の場合、 $nC = (2+3+1) >> 1 = 3$ となり、右から4番目のテーブルが使用される。一番右のテーブルの選択基準が $nC = -1$ となっており、これは色差信号のDC値の符号化のみに用いられる。

【0048】

【表7】

TrailingOnes	TotalCoeff	0 <= nC < 2	2 <= nC < 4	4 <= nC < 8	8 <= nC	nC = -1
0	0	1	11	1111	0000 11	01
0	1	0001 01	0010 11	0011 11	0000 00	0001 11
1	1	01	10	1110	0000 01	1
0	2	0000 0111	0001 11	0010 11	0001 00	0001 00
1	2	0001 00	0011 1	0111 1	0001 01	0001 10
2	2	001	011	1101	0001 10	001
0	3	0000 0011 1	0000 111	0010 00	0010 00	0000 11
1	3	0000 0110	0010 10	0110 0	0010 01	0000 011
2	3	0000 101	0010 01	0111 0	0010 10	0000 010
3	3	0001 1	0101	1100	0010 11	0001 01
0	4	0000 0001 11	0000 0111	0001 111	0011 00	0000 10
1	4	0000 0011 0	0001 10	0101 0	0011 01	0000 011
2	4	0000 0101	0001 01	0101 1	0011 10	0000 010
3	4	0000 11	0100	1011	0011 11	0000 000
0	5	0000 0000 111	0000 0100	0001 011	0100 00	-
1	5	0000 0001 10	0000 110	0100 0	0100 01	-
2	5	0000 0010 1	0000 101	0100 1	0100 10	-
3	5	0000 100	0011 0	1010	0100 11	-
0	6	0000 0000 0111 1	0000 0011 1	0001 001	0101 00	-
1	6	0000 0000 110	0000 0110	0011 10	0101 01	-
2	6	0000 0001 01	0000 0101	0011 01	0101 10	-
3	6	0000 0100	0010 00	1001	0101 11	-
0	7	0000 0000 0101 1	0000 0001 111	0001 000	0110 00	-
1	7	0000 0000 0111 0	0000 0011 0	0010 10	0110 01	-
2	7	0000 0000 101	0000 0010 1	0010 01	0110 10	-
3	7	0000 0010 0	0001 00	1000	0110 11	-
0	8	0000 0000 0100 0	0000 0001 011	0000 1111	0111 00	-
1	8	0000 0000 0101 0	0000 0001 110	0001 110	0111 01	-
2	8	0000 0000 0110 1	0000 0001 101	0001 101	0111 10	-
3	8	0000 0001 00	0000 100	0110 1	0111 11	-
0	9	0000 0000 0011 11	0000 0000 1111	0000 1011	1000 00	-
1	9	0000 0000 0011 10	0000 0001 010	0000 1110	1000 01	-
2	9	0000 0000 0100 1	0000 0001 001	0001 010	1000 10	-
3	9	0000 0000 100	0000 0010 0	0011 00	1000 11	-
0	10	0000 0000 0010 11	0000 0000 1011	0000 0111 1	1001 00	-
1	10	0000 0000 0010 10	0000 0000 1110	0000 1010	1001 01	-
2	10	0000 0000 0011 01	0000 0000 1101	0000 1101	1001 10	-
3	10	0000 0000 0010 0	0000 0001 100	0001 100	1001 11	-
0	11	0000 0000 0001 111	0000 0000 1000	0000 0101 1	1010 00	-
1	11	0000 0000 0001 110	0000 0000 1010	0000 0111 0	1010 01	-
2	11	0000 0000 0010 01	0000 0000 1001	0000 1001	1010 10	-
3	11	0000 0000 0011 00	0000 0001 0000	0000 1100	1010 11	-
1	12	0000 0000 0001 010	0000 0000 0111 0	0000 0101 0	1011 01	-
2	12	0000 0000 0010 101	0000 0000 0110 1	0000 0110 1	1011 10	-
3	12	0000 0000 0010 000	0000 0000 1100	0000 1000	1011 11	-
0	13	0000 0000 0000 1111	0000 0000 0101 1	0000 0111 01	1100 00	-
1	13	0000 0000 0000 001	0000 0000 0101 0	0000 0111 1	1100 01	-
2	13	0000 0000 0001 001	0000 0000 0100 1	0000 0100 1	1100 10	-
3	13	0000 0000 0001 100	0000 0000 0110 0	0000 0110 0	1100 11	-
0	14	0000 0000 0000 1011	0000 0000 0011 1	0000 0010 01	1101 00	-
1	14	0000 0000 0000 1110	0000 0000 0010 11	0000 0011 00	1101 01	-
2	14	0000 0000 0000 1101	0000 0000 0011 0	0000 0010 11	1101 10	-
3	14	0000 0000 0001 000	0000 0000 0010 0	0000 0010 10	1101 11	-
0	15	0000 0000 0000 0111	0000 0000 0010 01	0000 0001 01	1110 00	-
1	15	0000 0000 0000 1010	0000 0000 0010 00	0000 0010 00	1110 01	-
2	15	0000 0000 0000 0001	0000 0000 0010 10	0000 0001 11	1110 10	-
3	15	0000 0000 0000 1100	0000 0000 0000 1	0000 0001 10	1110 11	-
0	16	0000 0000 0000 0100	0000 0000 0001 11	0000 0000 01	1111 00	-
1	16	0000 0000 0000 0010	0000 0000 0001 10	0000 0000 01	1111 01	-
2	16	0000 0000 0000 0001	0000 0000 0001 01	0000 0000 11	1111 10	-
3	16	0000 0000 0000 1000	0000 0000 0001 00	0000 0000 10	1111 11	-

表7 TrailingOnesとTotalCoeffのVLCテーブル

【0049】

直交変換係数値(level)はレベル符号化回路4に供給されて可変長符号化される。レベル符号化回路4では、直交変換係数値から level_prefix および level suffix と呼ばれるパラメータを抽出する。

level_prefix は以下に示すVCLテーブルを用いて可変長符号化する。

【0050】

【表8】

level_prefix	Bit String
0	1
1	01
2	001
3	0001
4	0000 1
5	0000 01
6	0000 001
7	0000 0001
8	0000 0000 1
9	0000 0000 01
10	0000 0000 001
11	0000 0000 0001
12	0000 0000 0000 1
13	0000 0000 0000 01
14	0000 0000 0000 001
15	0000 0000 0000 0001

表8 level_prefixの復号VLCテーブル

【0051】

level_suffix は suffixLength で与えられるビット長で unsigned integer として符号化される。

level と level_prefix, level_suffix の関係は以下の式で与えられる。

```
levelCode = (level_prefix << suffixLength) + level_suffix
```

levelCodeが偶数の場合 : level = (levelCode + 2) >> 1

levelCodeが偶数でない場合 : level = (-levelCode - 1) >> 1

【0052】

ラン (run_before) およびランの総和 (total_zeros) はラン符号化回路3に供給され可変長符号化される。以下にはランの総和 (total_zeros)、および、ラン (run_before) に用いられる VLC テーブルを示す。

【0053】

【表9】

total_zeros	TotalCoeff						
	1	2	3	4	5	6	7
0	1	111	0101	00011	0101	000001	000001
1	011	110	111	111	0100	00001	00001
2	010	101	110	0101	0011	111	101
3	0011	100	101	0100	111	110	100
4	0010	011	0100	110	110	101	011
5	00011	0101	0011	101	101	100	11
6	00010	0100	100	100	100	011	010
7	000011	0011	011	0011	011	010	001
8	000010	0010	0010	011	0010	0001	001
9	0000011	00011	00011	0010	00001	001	00000
10	0000010	00010	00010	00010	0001	00000	
11	00000011	000011	000001	00001	00000		
12	00000010	000010	00001	00000			
13	000000011	000001	000000				
14	000000010	000000					
15	000000001						

表9 4x4ブロックのtotal_zeros復号用VLCテーブル (1<=TotalCoeff <=7)

【0054】

【表10】

total_zeros	TotalCoeff							
	8	9	10	11	12	13	14	15
0	000001	000001	00001	0000	0000	000	00	0
1	0001	000000	00000	0001	0001	001	01	1
2	00001	0001	001	001	01	1	1	
3	011	11	11	010	1	01		
4	11	10	10	1	001			
5	10	001	01	011				
6	010	01	0001					
7	001	00001						
8	000000							

表10 4x4ブロックのtotal_zeros復号用VLCテーブル (8<=TotalCoeff <=15)

【0055】

【表11】

total_zeros	TotalCoeff		
	1	2	3
0	1	1	1
1	01	01	0
2	001	00	
3	000		

表11 色差DC 2x2ブロックのtotal_zeros復号用VLCテーブル

【0056】

【表12】

run_before	zerosLeft						
	1	2	3	4	5	6	>6
0	1	1	11	11	11	11	111
1	0	01	10	10	10	000	110
2	-	00	01	01	011	001	101
3	-	-	00	001	010	011	100
4	-	-	-	000	001	010	011
5	-	-	-	-	000	101	010
6	-	-	-	-	-	100	001
7	-	-	-	-	-	-	0001
8	-	-	-	-	-	-	00001
9	-	-	-	-	-	-	000001
10	-	-	-	-	-	-	0000001
11	-	-	-	-	-	-	00000001
12	-	-	-	-	-	-	000000001
13	-	-	-	-	-	-	0000000001
14	-	-	-	-	-	-	00000000001

表12 ラン(run_before)復号用VLCテーブル

【0057】

可変長符号されたこれらの情報は多重化回路5により1本のビットストリームに多重化され出力される。

表13にTRSIZEをビットストリーム中に符号化する際のシンタクスの一例を示す。表13における、transform_size_8x8_flagがTRSIZEである。この値が0の場合、4x4ブロックサイズの直交変換が適用され、この値が1の場合、8x8ブロックサイズの直交変換が適用される。

【0058】

【表 13】

Macroblock layer syntax

macroblock_layer() {	C	Descriptor
mb_type	2	ue(v) ae(v)
if(mb_type == I_PCM) {		
while(!byte_aligned())		
pcm_alignment_zero_bit	2	f(1)
for(i = 0; i < 256; i++)		
pcm_sample_luma[i]	2	u(v)
for(i = 0; i < 256 * (ChromaFormatFactor - 1); i++)		
pcm_sample_chroma[i]	2	u(v)
} else {		
if(mb_type != I_NxN &&		
MbPartPredMode(mb_type, 0) != Intra_16x16 &&		
NumMbPart(mb_type) == 4) {		
sub_mb_pred(mb_type)	2	
NoSubMbPartSizeLessThan8x8Flag = 1		
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8)		
if(NumSubMbPart(sub_mb_type[mbPartIdx]) > 1)		
NoSubMbPartSizeLessThan8x8Flag = 0		
else		
NoSubMbPartSizeLessThan8x8Flag = 0		
if(NoSubMbPartSizeLessThan8x8Flag &&		
transform_8x8_mode_flag)		
transform_size_8x8_flag	2	ae(v)
} else {		
if(MbPartPredMode(mb_type, 0) != Intra_16x16) {		
if(transform_8x8_mode_flag &&		
!(MbPartPredMode(mb_type, 0) == B_Direct_16x16		
&& ldirect_8x8_inference_flag))		
transform_size_8x8_flag	2	ae(v)
mb_pred(mb_type)	2	
}		
}		
if(MbPartPredMode(mb_type, 0) != Intra_16x16)		
coded_block_pattern	2	mc(v) ac(v)
if(CodedBlockPatternLuma > 0 CodedBlockPatternChroma > 0		
MbPartPredMode(mb_type, 0) == Intra_16x16) {		
mb_qp_delta	2	sc(v) ae(v)
residual()	3 4	
}		
}		
}		

表 13 マクロブロックレイヤのシンタクス

【0059】

図10に離散コサイン変換若しくはカルーネン・レーベ変換等の直交変換と動き補償により画像圧縮を実現する画像情報復号化装置のブロック図を示す。すなわち、13は蓄積バッファ、14は可逆符号化装置、15は逆量子化装置、16は逆直交変換装置、17は加算器、18は画面並べ替えバッファ、19は、D/A変換装置、20は動き予測・補償装置、21はフレームメモリ、22はイントラ予測装置である。

図10において、入力となる画像圧縮情報は、まず、蓄積バッファ13に格納された後、可逆復号化装置14に転送される。ここで、定められた画像圧縮情報のフォーマットに基づき、可変長復号化、算術復号化等の処理が行われる。同時に、当該フレームがインターフォーマット化されたものである場合には、可逆復号化装置においては、画像圧縮情報のヘッダ部に格納された動きベクトル情報をも復号化し、その情報を動き予測・補償装置20へ転送する。

送する。

可逆復号化装置の出力となる、量子化された変換係数は、逆量子化装置15に入力され、ここで変換係数として出力される。変換係数は、逆直交変換装置16において、定められた画像圧縮情報のフォーマットに基づき、逆離散コサイン変換、逆カルーネン・レーベ変換等の逆直交変換が施される。当該フレームがイントラ符号化されたものである場合には、逆直交変換処理が施された画像情報は画面並べ替えバッファに格納され、D/A変換処理の後に出力される。

【0060】

当該フレームがインター符号化されたものである場合には、可逆復号化処理が施された動きベクトル情報、及びフレームメモリ21に格納された画像情報を元に参照画像が生成され、この参照画像と、逆直交変換装置16の出力とが、加算器17において合成される。その他の処理はイントラ符号化されたフレームと同様である。

当該フレームがイントラ符号化されたものである場合には、可変長復号されたイントラ予測モードに従い、フレームメモリから画像を読み出し、イントラ予測装置22により予測画像が生成され、この予測画像と、逆直交変換装置16との出力が加算器7において合成される。

本実施例では可逆符号復号装置14はビットストリーム中に符号化され伝送されてくる、直交変換のブロックサイズを示すフラグ TRSIZE を復号し、このフラグに基づき、直交変換係数を復号し、逆量子化装置15に出力する。また、復号した TRSIZE を逆量子化装置15および逆直交変換装置16に供給する。

【0061】

TRSIZE が 4×4 ブロックサイズの直交変換であることを示す場合、図11に示すような構成で、可逆符号復号装置14は直交変換係数を復号する。ビットストリームはまず逆多重化回路1に供給され、符号化されている。

レベル (level, 係数値)

ラン (run_before, 非0係数値の前の連続する0係数の数)

ランの総和 (total_zeros, 最後の非係数以前の0係数の数)

非0係数の個数 (TotalCoeff)

最後に連続する絶対値1の係数の個数 (TrailingOnes)

最後に連続する絶対値1の係数の符号 (trailing_ones_sign_flag)

をそれぞれ、ラン復号回路2、レベル復号回路3、2次元可逆符号復号回路6に供給される。ラン復号回路2では、表9～表12の可変長符号のテーブルを用い、可変長符号を復号し、ラン (run_before) およびランの総和 (total_zeros) を変換係数復元回路4に供給する。

【0062】

レベル復号回路3では、表8の可変長符号テーブルを用いて各係数のレベル値を復号し、変換係数復元回路4に供給する。

2次元可逆符号復号回路6では、表7の可変長符号テーブルを用いて、非0係数の個数 (TotalCoeff) と最後に連続する絶対値1の係数の個数 (TrailingOnes) を復号し、変換係数復元回路4に供給する。この時、nC を計算する方式は符号化装置と同じ方式を用いる。

【0063】

変換係数復元回路4では、

レベル (level, 係数値)

ラン (run_before, 非0係数値の前の連続する0係数の数)

ランの総和 (total_zeros, 最後の非係数以前の0係数の数)

非0係数の個数 (TotalCoeff)

最後に連続する絶対値1の係数の個数 (TrailingOnes)

最後に連続する絶対値1の係数の符号(trailing_ones_sign_flag)

から16個の直交変換係数を復元し、スキャン変換回路5に供給する。

【0064】

スキャン変換回路5では、1次元の直交変換係数を図4のスキャン順に基づいて2次元の変換係数に復元する。復元された直交変換係数は、逆量子化装置に出力される。

TRSIZEx8x8ブロックサイズの直交変換であることを示す場合、図12に示すような構成で、可逆符号復号装置14は直交変換係数を復号する。ビットストリームはまず逆多重化回路1に供給され、符号化されている。

【0065】

レベル (level, 係数値)

ラン (run_before, 非0係数値の前の連続する0係数の数)

ランの総和 (total_zeros, 最後の非係数以前の0係数の数)

非0係数の個数 (TotalCoeff)

最後に連続する絶対値1の係数の個数(TrailingOnes)

最後に連続する絶対値1の係数の符号(trailing_ones_sign_flag)

をそれぞれ、ラン復号回路2、レベル復号回路3、2次元可逆符号復号回路6に供給される。ラン復号回路2では、表9～表12の可変長符号のテーブルを用い、可変長符号を復号し、ラン (run_before) およびランの総和 (total_zeros) を変換係数復元回路4に供給する。

【0066】

レベル復号回路3では、表8の可変長符号テーブルを用いて各係数のレベル値を復号し、変換係数復元回路4に供給する。

2次元可逆符号復号回路6では、表7の可変長符号テーブルを用いて、非0係数の個数(TotalCoeff)と最後に連続する絶対値1の係数の個数(TrailingOnes)を復号し、変換係数復元回路4に供給する。この時、nCを計算する方式は符号化装置と同じ方式を用いる。

【0067】

変換係数復元回路4では、

レベル (level, 係数値)

ラン (run_before, 非0係数値の前の連続する0係数の数)

ランの総和 (total_zeros, 最後の非係数以前の0係数の数)

非0係数の個数 (TotalCoeff)

最後に連続する絶対値1の係数の個数(TrailingOnes)

最後に連続する絶対値1の係数の符号(trailing_ones_sign_flag)

から16個の直交変換係数を復元し、ブロック復元回路6に入力する。

【0068】

ブロック復元回路6では、図9のサブブロック1に相当する最初の16個の係数、サブブロック2に相当する次の16個の係数、サブブロック3に相当する次の16個の係数、サブブロック4に相当する最後の16個の係数の順で変換係数をスキャン変換回路5に供給する。

スキャン変換回路5では、1次元の直交変換係数を図13のスキャン順に基づいて2次元の変換係数に復元する。復元された直交変換係数は、逆量子化装置に出力される。

【産業上の利用可能性】

【0069】

本発明は、本発明は、MPEG、H.26x等のシステムに適用可能である。

【図面の簡単な説明】

【0070】

【図1】本発明の関連技術を説明するための図である。

【図2】本発明の関連技術を説明するための図である。

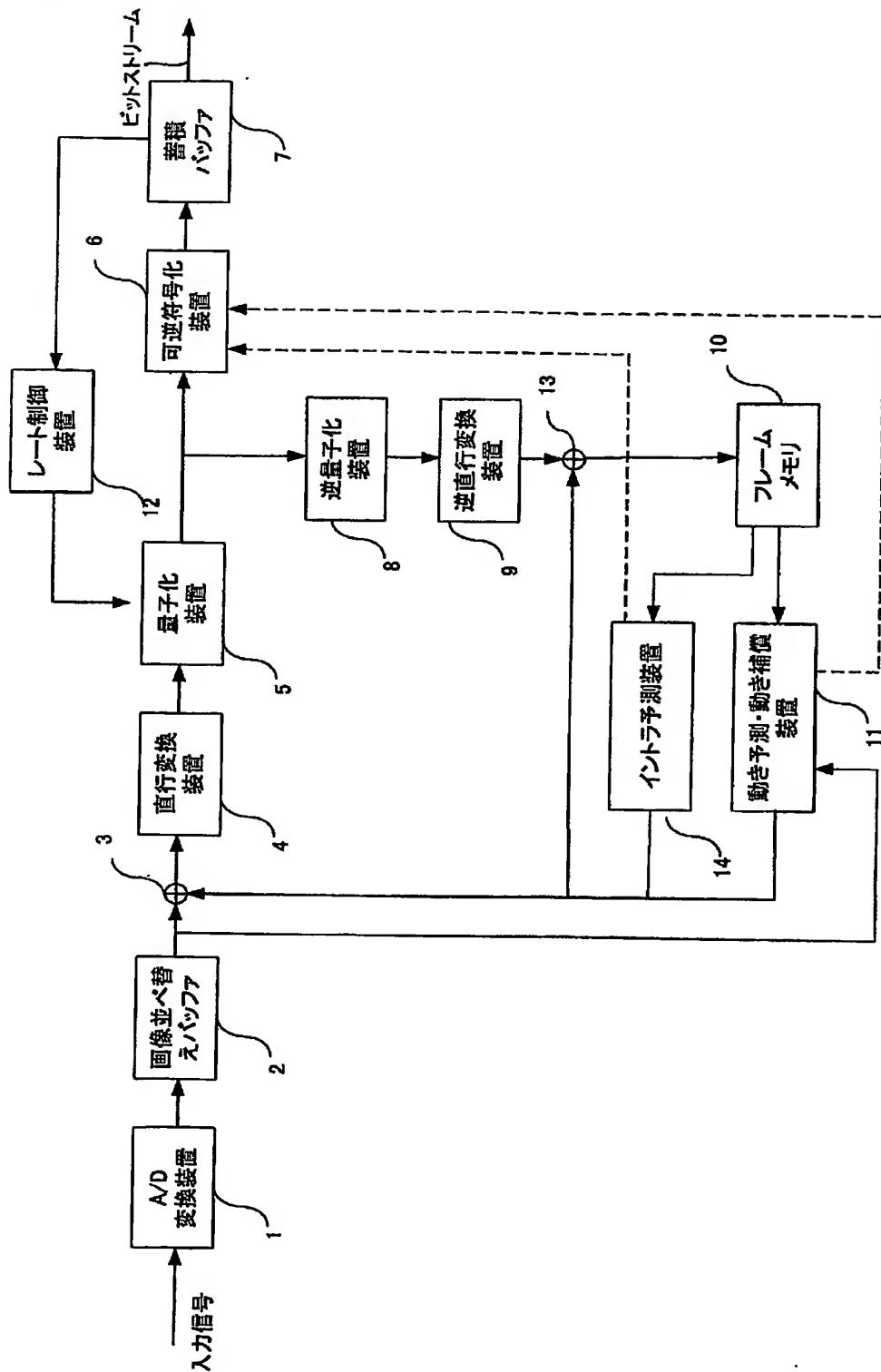
【図3】本発明の関連技術を説明するための図である。
【図4】本発明の関連技術を説明するための図である。
【図5】本発明の関連技術を説明するための図である。
【図6】本発明の実施形態を説明するための図である。
【図7】本発明の実施形態を説明するための図である。
【図8】本発明の実施形態を説明するための図である。
【図9】本発明の実施形態を説明するための図である。
【図10】本発明の実施形態を説明するための図である。
【図11】本発明の実施形態を説明するための図である。
【図12】本発明の実施形態を説明するための図である。
【図13】本発明の実施形態を説明するための図である。

【符号の説明】

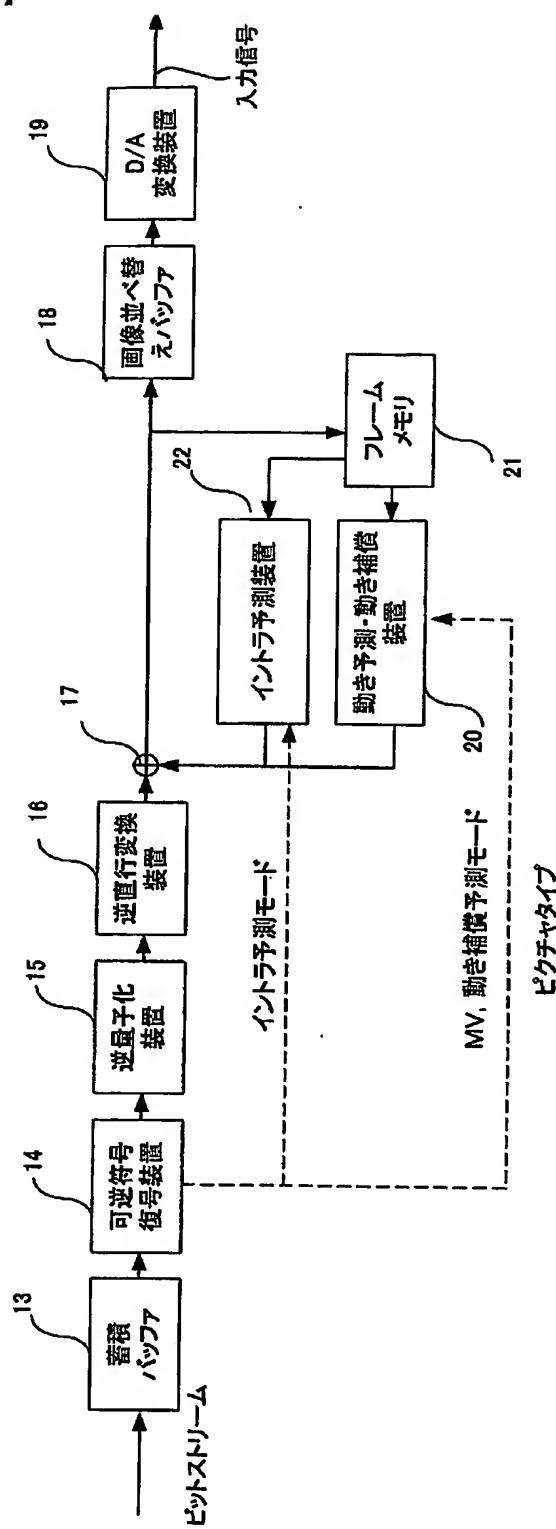
【0071】

1…スキャン変換回路、2…ラン、レベル計算回路、3…ラン符号化回路、4…レベル
符号化回路、5…多重化回路、6…サブブロック分割回路、7…2次元可逆符号化回路

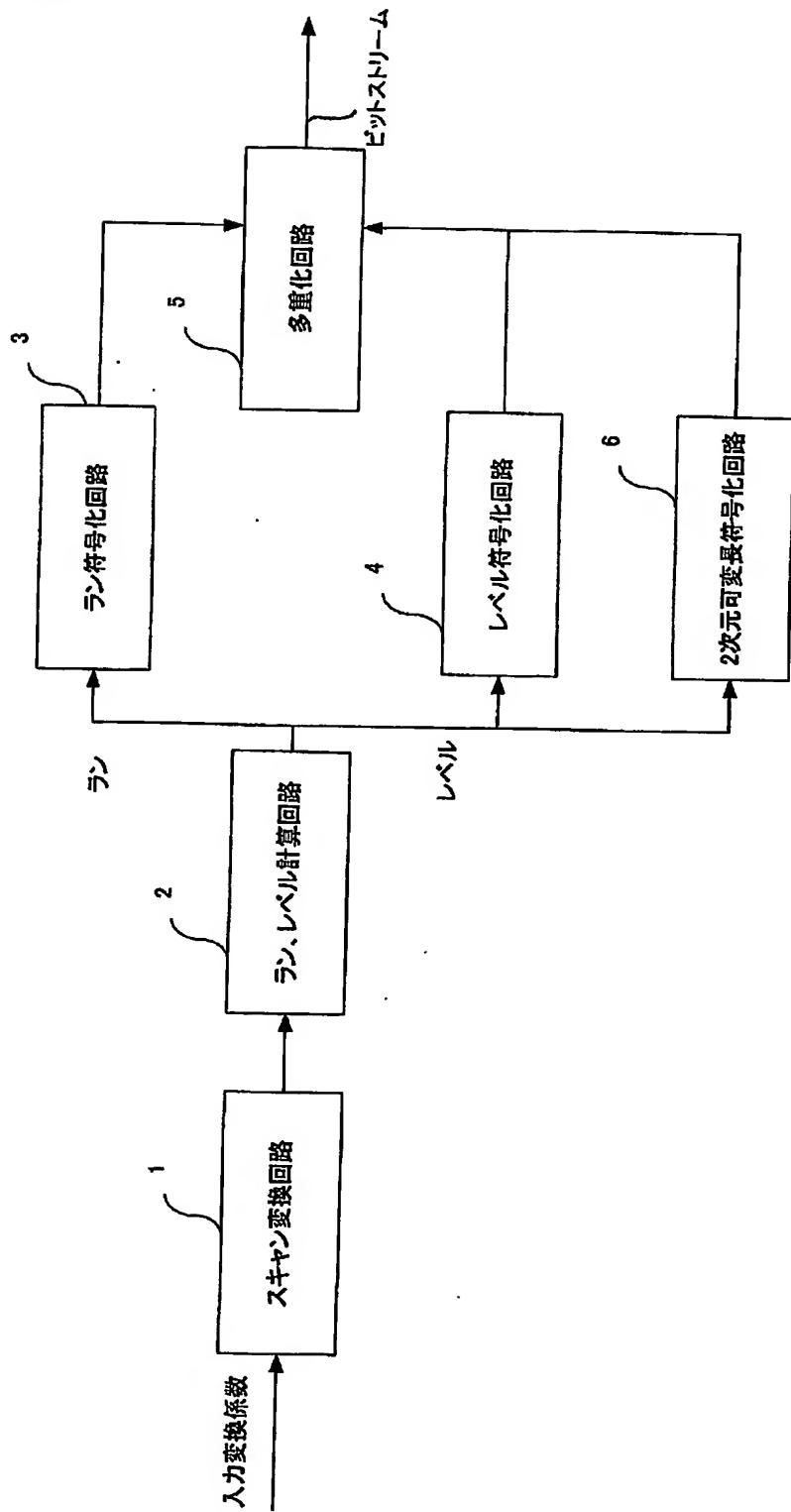
【書類名】 図面
【図 1】



【図2】



【図3】



【図4】

0	1	5	6
2	4	7	12
3	8	11	13
9	10	14	15

(a)

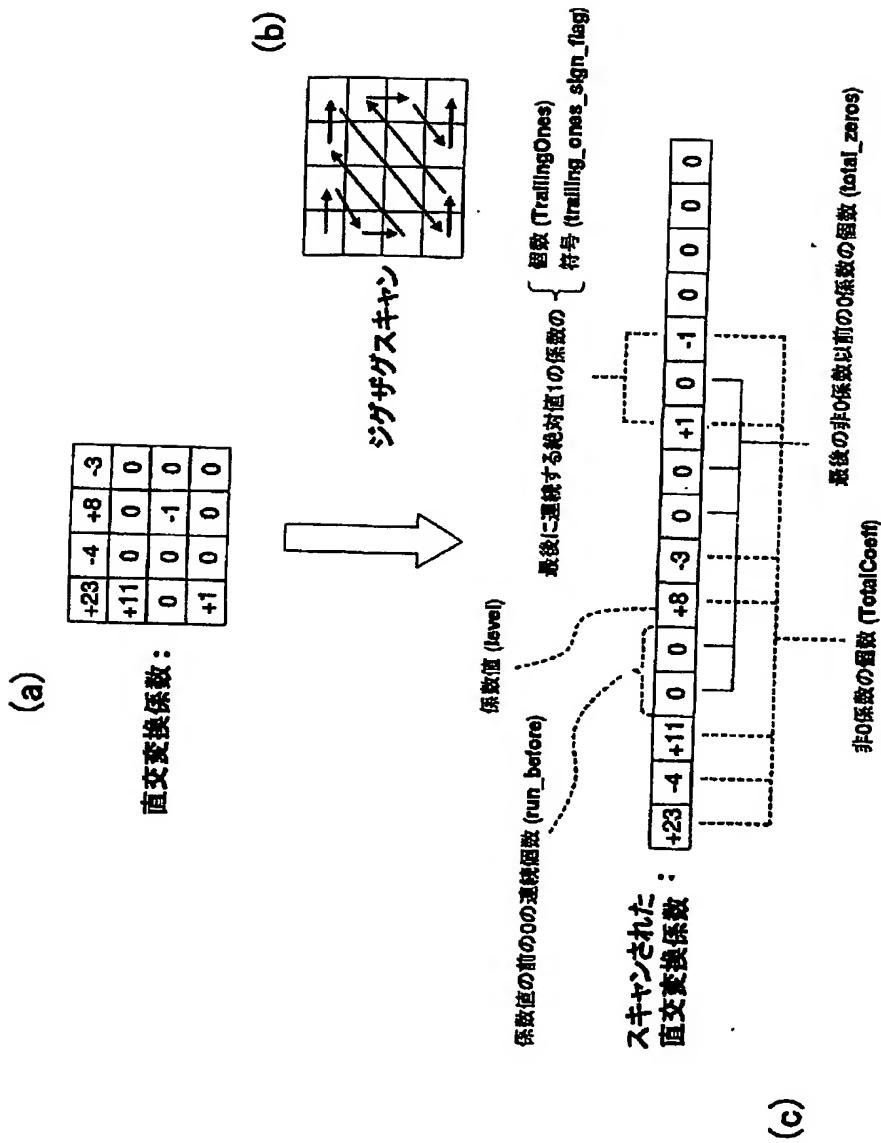
0	2	8	12
1	5	9	13
3	6	10	14
4	7	11	15

(b)

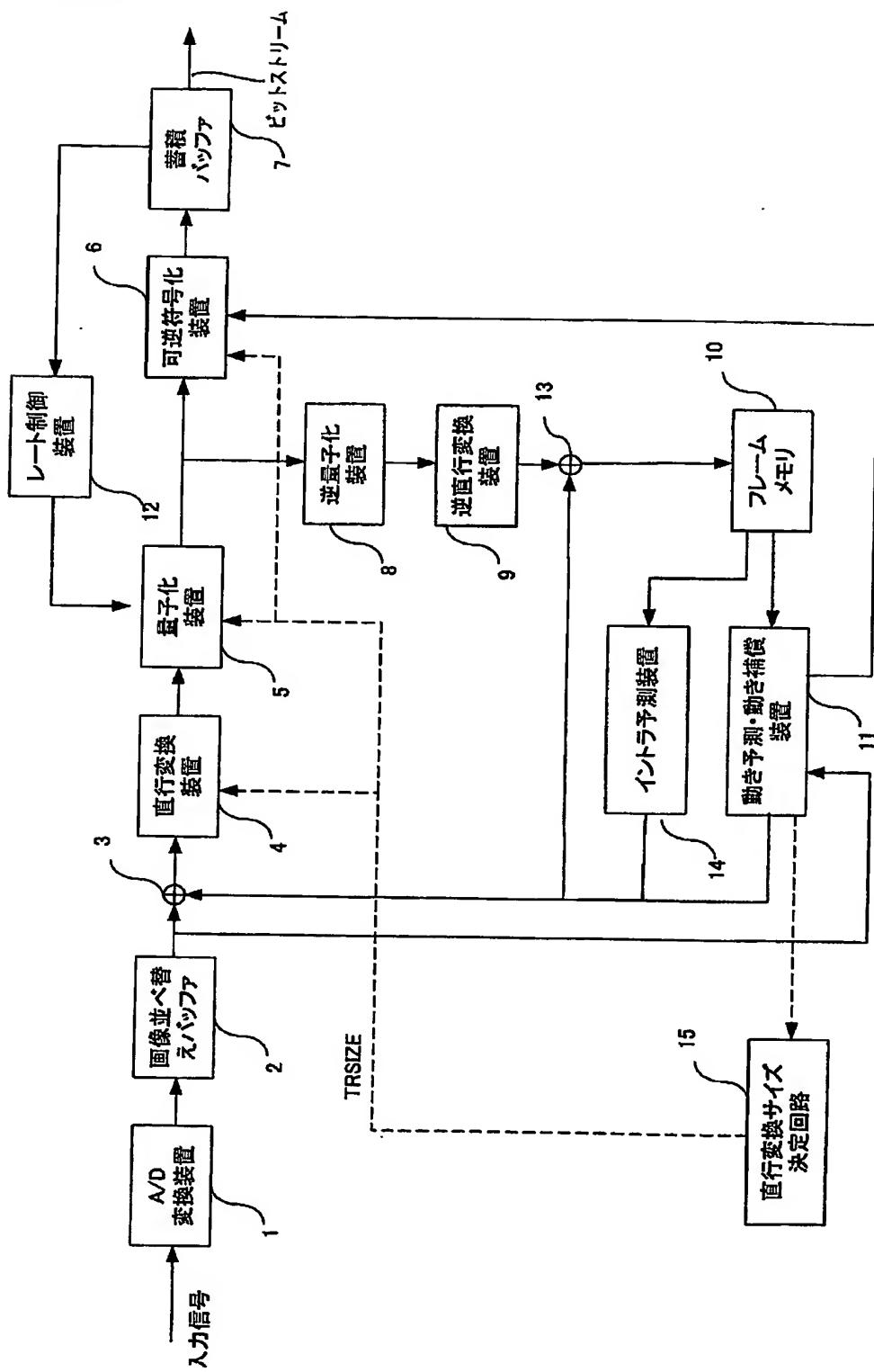
ジグザグスキャン

フィールドスキャン

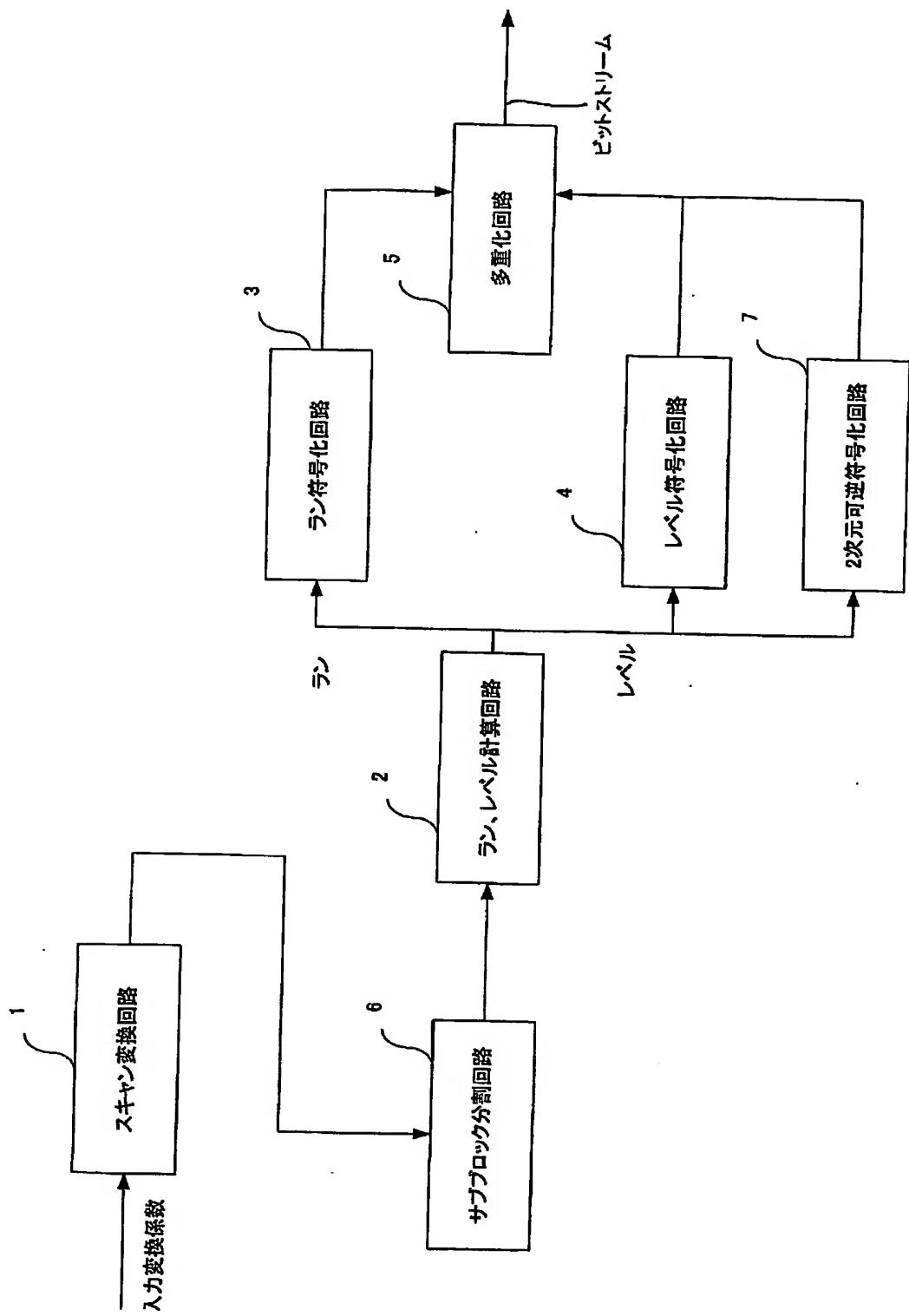
【図 5】



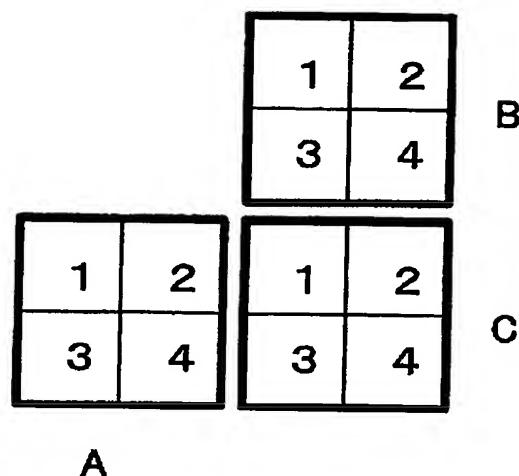
【図6】



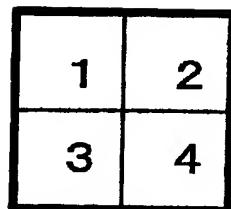
【図 7】



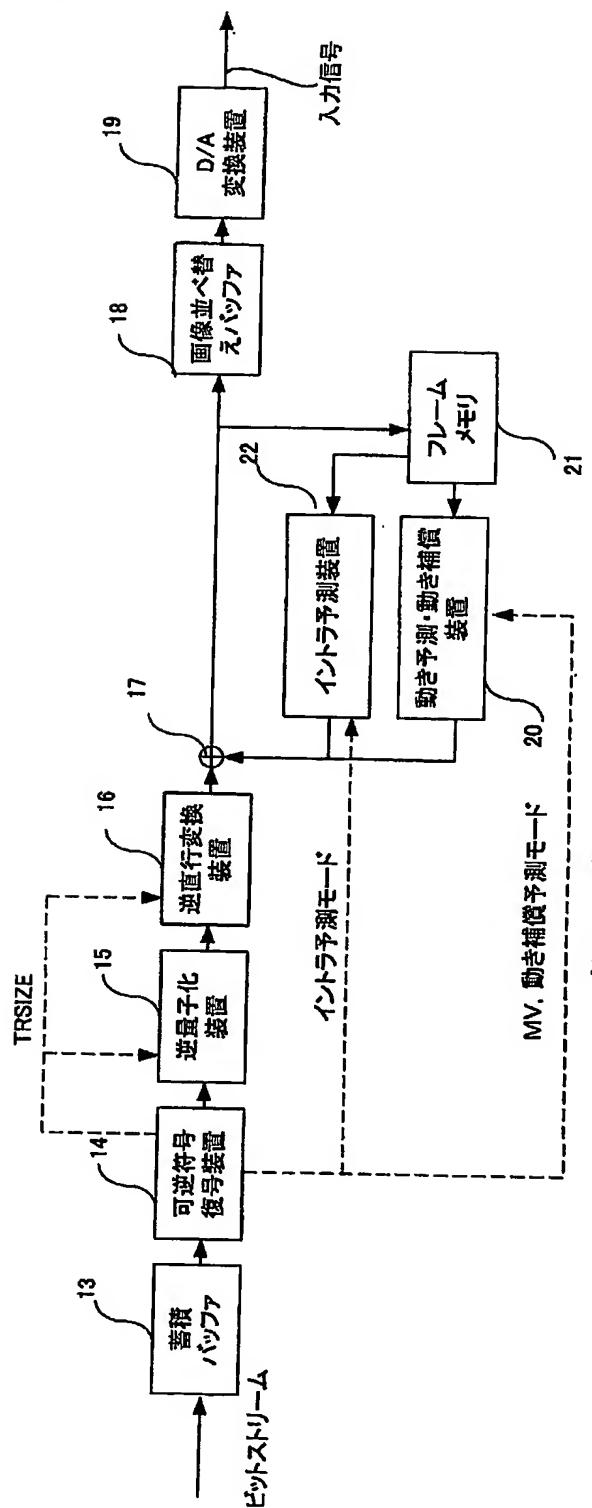
【図8】



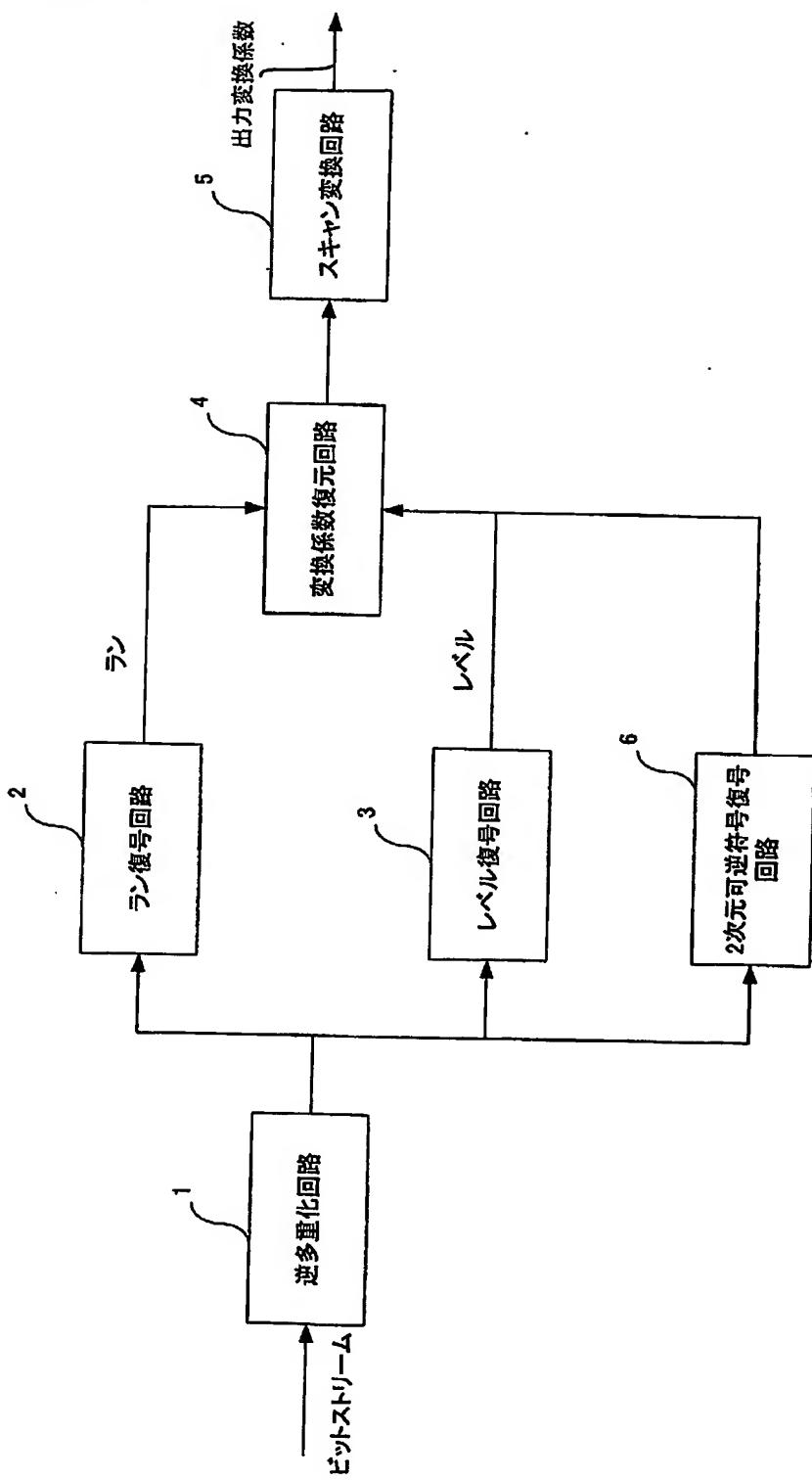
【図9】



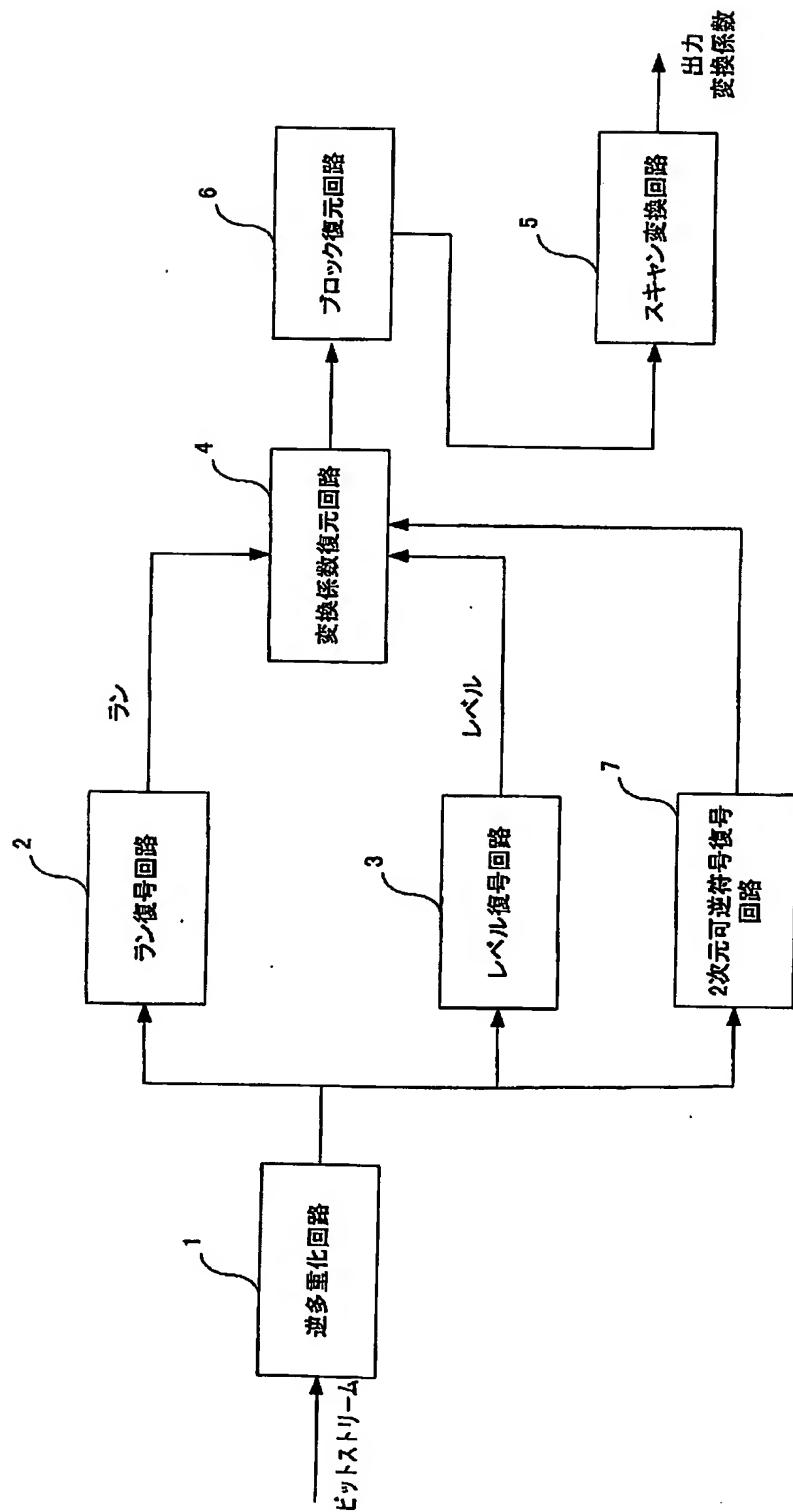
【図10】



【図11】



【図12】



【図13】

		u							
		0	1	2	3	4	5	6	7
0		0	1	5	6	14	15	27	28
1		2	4	7	13	16	26	29	42
2		3	8	12	17	25	30	41	43
3		9	11	18	24	31	40	44	53
4		10	19	23	32	39	45	52	54
5		20	22	33	38	46	51	55	60
6		21	34	37	47	50	56	59	61
v		35	36	48	49	57	58	62	63

【書類名】 要約書

【要約】

【課題】 8×8 , 4×4 など異なるブロックサイズの直交変換を行った際に効率よく可変長符号化を実現する。

【解決手段】 直交変換のブロックサイズを示すフラグに応じ、大きいブロックサイズ（例えば 8×8 ブロック）の場合、小さいブロックサイズ（例えば 4×4 ブロック）に適応的に再分割し、小さいブロックサイズでの可変長符号化をそれぞれの再分割したブロックに適応することにより、より効率的に可変長符号化を実現することを特徴とする。

【選択図】 図 7

特願 2004-204313

出願人履歴情報

識別番号

[000002185]

1. 変更年月日
[変更理由]

1990年 8月30日

新規登録

住所
氏名
東京都品川区北品川6丁目7番35号
ソニー株式会社